IN THE SPECIFICATION

Please amend the specification as provided below:

The present invention now will be described more fully with reference to the accompanying drawings, in which the preferred embodiments of the invention are shown. The present invention may, however, be embodied in many different forms and should not be construed as limited to the embodiment set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete and will fully convey the invention to those skilled in the art.

Please amend the paragraph beginning on page 7, line 21 as follows:

Fig. 2 also shows that the buffer manager 212 is directly connected via a separate data interface 218 to a flash memory sequencer 220. Under program control, the microcontroller 216 directs the buffer manager 212 to sequentially move both the ~~each~~ data segment or sector stored in buffer 214 through a FIFO like data register (first-in/first-out) of the buffer manager ~~and buffer~~ 212 and across the attached data interface 218 to the flash memory sequencer 220. Upon receipt of the two data strings by flash memory sequencer 220, an ECC error correction procedure is performed prior to being processed and written to flash memory 222. This allows errors that would normally cause a problem, to be detected and corrected without effecting the operation of the system. Once the ECC error correction process is complete, the flash memory sequencer 220 then transfers ~~the~~ both the odd and even data segments as well as the associated error correction code via a flash memory data interface 224 to flash memory module 222a and flash memory module 222b, respectfully.

Please amend the paragraph beginning on page 8, line 1 as follows:

When data is read from flash memory 222, the requested odd and even data segments are transferred from flash memory module 222a and flash memory module 222b, respectfully across the flash memory data interface 224 to the flash memory sequencer 220. The data segments are then moved to the buffer 214 of the buffer manager ~~and~~ 212 where they are concatenated into a complete data word that can be transferred back to the host either through the PCMCIA-ATA interface 203 or the IDE Interface 204.

Please amend the paragraph beginning on page 8, line 12 as follows:

~~FIG. 3 is a flowchart~~ FIGs. 3a-3i are exemplary flowcharts that illustrates the flow of events performed by the compact flash controller in accordance with FIG. 2. The steps in the flowcharts are simply illustrative of the functional steps performed by the ~~by the~~ compact flash controller 200, however, a person of ordinary skill in the art will appreciate that the exact sequence of operations by the compact flash controller 200 to perform the functions described in the flowchart of ~~FIG. 3~~ FIGs. 3a-3i may vary. Reference is now to ~~Figs. 3a~~ Fig. 3a of flowchart illustrating the steps performed by the compact flash controller to manage data transfers in and out of flash memory 222. As Fig. 3a shows at steps 302, the card is inserted and detected, while at step 304, all components of the card including the compact flash controller 200, are powered up and initialized. At step 308, the flash memory 222, is then initialized and partitioned.

Please amend the paragraph beginning on page 8, line 23 as follows:

Fig. 3b is a flow chart that further details the steps used by the compact flash controller to initialize and partition the plurality of installed flash memory modules. As Fig. 3b shows, step 308 further includes at step 324 the flash memory module or arrays that are installed on the compact flash memory card inserted into the host device are

initialized. In addition, after the manufacture and device codes are detected and scanned, at step 325, compact flash memory controller, at step 326, detects the number of flash memory modules or arrays present on the card. The compact flash memory controller, at step ~~327~~ 328, then determines if the flash memory has been partitioned in accordance with the storage requirements of the host device. If the flash memory has been partitioned, the compact flash memory controller~~, at step 329,~~ continues on to step 310 that determines which interface is to be used. On the other hand, if the flash memory has not been partitioned, the compact flash controller, at step ~~328~~ 329, partitions the flash memory in accordance with the configuration information structure (CIS) stored in the read only memory (ROM), together with other software programs. Thus, when a flash memory card is inserted into the card slot of the host device, the host computer searches for the configuration information structure (CIS) of this flash memory card. In the flash memory card, the controller reads the CIS information from the ROM and places it in RAM or a register that the host computer can access. Therefore, in accordance with operative elements of the configuration information structure (CIS), the host computer then assigns memory space, I/O space area, interrupt level and sequential read/write-accesses to the flash memory.

Please amend the paragraph beginning on page 9, line 14 as follows:

Referring now to Fig. 3c that details the operative elements of ~~step328~~ step 329 performed by the compact flash controller to configure and partition the plurality of flash memory modules or arrays installed on the installed compact flash memory card. As Fig. 3c shows, the compact flash controller, at step 331, locates the last data block in each flash memory module. If the block is not defective, the controller designates it for use as a bad block reference table. Each bit in the bad block reference table indicates what blocks of the flash memory module or array is either "good or bad". Then, at step 332, the compact flash controller designates the next available data block, if not defective, to be used as a drive information table. The drive information table maintains